Assignment2 - Step 7

Phoebe Zhou

September 2025

Final Summary Report

1. Introduction

Retrieval-Augmented Generation (RAG) offers a powerful way to combine the factual reliability of external knowledge sources with the generative flexibility of language models. This project set out to explore how different design decisions in a RAG pipeline affect performance, starting from a simple baseline and gradually introducing enhancements such as query rewriting and reranking.

The dataset was a Wikipedia-derived collection that paired around 918 factual questions with roughly 3,200 supporting passages. Both the questions and answers were typically short and direct (often yes/no or single-word facts), making precision and faithfulness especially critical.

The following report summarizes the design of the naive baseline system, the results of experimentation, the rationale for enhancements, automated evaluation outcomes, and reflections on lessons learned, limitations, and future directions.

2. The Naive RAG System

The naive pipeline followed the standard RAG blueprint but in its simplest form. Passages from the corpus were embedded using a lightweight MiniLM model and stored in a FAISS index to enable fast nearest-neighbor search. At query time, a question was embedded and compared against the corpus, and the single most similar passage was retrieved. This top-1 passage was then fed, alongside the question, into a flan-t5-base generator model to produce an answer.

Prompting proved to be a key factor even at this baseline stage. I experimented with three templates: a direct instruction prompt, a persona-style prompt that framed the model as a helpful tutor, and a chain-of-thought prompt encouraging step-by-step reasoning. While the mechanics of the pipeline were identical in each case, the tone and structure of the answers differed significantly.

3. Early Experiments: Prompting and Retrieval Variations

Evaluation with Exact Match and F1 scores revealed stark differences in performance across prompting strategies. Instruction prompting consistently outperformed the others, with F1 around 39 and Exact Match around 32. Persona prompting scored lower, since its conversational style strayed from the concise ground truth answers, and chain-of-thought prompting performed worst, with verbose reasoning that rarely aligned with the short factual references. This confirmed that in a dataset dominated by short, fact-based QA, clarity and conciseness in prompts are crucial.

I then turned attention to retrieval parameters. Different MiniLM embedding models were tested, alongside variations in retrieval depth (k=3, 5, 10) and strategies for handling multiple retrieved passages. Concatenating the top-10 passages produced clear gains, pushing F1 above 42, while relying only on the top passage often left the model without sufficient evidence.

Interestingly, embedding size mattered less than expected: the smaller 256-dimensional variant performed competitively with its larger 384-dimensional counterparts, suggesting efficiency could be traded for only a modest drop in accuracy. A heuristic approach that selected passages by length similarity performed poorly, underscoring that semantic relevance cannot be replaced by shortcuts.

4. Moving Beyond the Baseline

The experiments highlighted two main weaknesses of the naive pipeline: retrieval noise when the top passage was off-target, and occasional hallucination or irrelevance from the generator when evidence was thin. To address these, I introduced two enhancements. First, a query rewriter expanded or clarified questions before embedding, helping to surface more relevant passages. Second, a reranker was applied to reorder retrieved passages by likelihood of containing the correct answer, rather than relying solely on embedding similarity. These changes aimed directly at improving the quality of context passed to the generator.

5. Automated Evaluation with RAGAs

To rigorously compare the naive and enhanced pipelines, I applied RAGAs, which provides metrics focused on grounding and factual accuracy: faithfulness, answer relevancy, context precision, and context recall. Because each evaluation involves many model calls, I limited the run to 200 queries per system rather than the full 918. While this sampling does not capture every possible variation, it provides a reliable picture of relative performance.

The results showed the enhanced pipeline clearly outperformed the naive one. Faithfulness rose from 0.86 to 0.94, meaning the enhanced model was more likely to stick closely to the

retrieved evidence. Answer relevancy improved slightly, reflecting that both systems already produced mostly relevant outputs, but the enhanced version was marginally better. The largest jumps were in context precision and recall: precision climbed from 0.69 to 0.85, while recall improved from 0.66 to 0.80. These gains highlight the value of query rewriting and reranking in filtering out noise and ensuring that more of the correct evidence is surfaced. For a dataset where answers are often just a word or two, such improvements are critical.

6. Lessons Learned

One of the clearest lessons from this work is that prompting style can have as much impact as backend architecture. Direct instruction kept answers concise and aligned with the gold standard, while stylistic prompts consistently hurt performance. Another insight is that retrieval quality often matters more than embedding size. Adding more context through top-k concatenation produced larger gains than switching between small and medium embedding models. The enhancements confirmed that improving retrieval precision and recall translates directly into better system outputs.

7. Limitations

Despite these advances, several limitations remain. The dataset itself is narrow and fact-oriented, which means results may not generalize to more complex, open-ended tasks. Automated metrics like Exact Match are strict and penalize answers that are semantically correct but phrased differently, while RAGAs sometimes struggled to parse outputs. Resource constraints also shaped the evaluation: larger models might improve results further, but they were not practical within this project's limits, and only 200 queries could be evaluated with RAGAs due to time costs. Finally, while query rewriting and reranking improved retrieval, they introduced added computational overhead.

8. Conclusion

This project demonstrated the full process of building and refining RAG pipelines, from a naive baseline to an enhanced system with query rewriting and reranking. The naive pipeline proved surprisingly competent for straightforward fact-based questions but was undermined by retrieval errors and occasional hallucinations.

Through systematic experimentation, it became clear that precision in retrieval and clarity in prompting were the main levers for improvement. The enhanced pipeline leveraged these insights to deliver consistent gains across all key metrics, particularly in context precision and recall, which are vital for fact-grounded QA. While the evaluation was limited to 200 queries per system, the trends were unmistakable: enhancements produced more accurate,

faithful, and trustworthy answers. The work illustrates how even small, targeted changes can significantly improve RAG performance and lays the groundwork for further research into smarter retrieval and evaluation methods.